NAVIWAM: COUPLED WORLD–ACTION PREDICTION FOR EMBODIED NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Embodied navigation requires agents to translate task specifications and evolving visual observations into executable motion. Action labels supervise low-dimensional trajectories but do not directly constrain the richer visual changes along a route. We introduce NaviWAM, a coupled world–action prediction framework that complements trajectory supervision with patch-level next-frame prediction. A vision–language model integrates the task with long-horizon visual history to condition a shared dual-stream predictor. The predictor supports two separately invoked tasks: trajectory prediction (Policy) and next-frame feature prediction (NF). Shared parameters and cross-stream attention allow NF supervision to update components used for navigation. Labeled navigation sequences supervise both tasks, while additional video–instruction pairs provide NF targets without trajectory annotations. The same objective supports offline target-scene adaptation: target videos contribute NF supervision, while source trajectories retain Policy supervision through replay. At deployment, only the Policy task is sampled, without generating future visual observations. With separately trained vision-and-language and point-goal models, NaviWAM achieves competitive results across continuous navigation benchmarks. The combined Policy + NF training scheme increases RxR-CE success rate from 58.9% to 63.6%. On VLNVerse, target-scene NF adaptation improves fine- and coarse-grained success rates by 6.91 and 7.06 percentage points, respectively, without target-scene trajectory supervision.

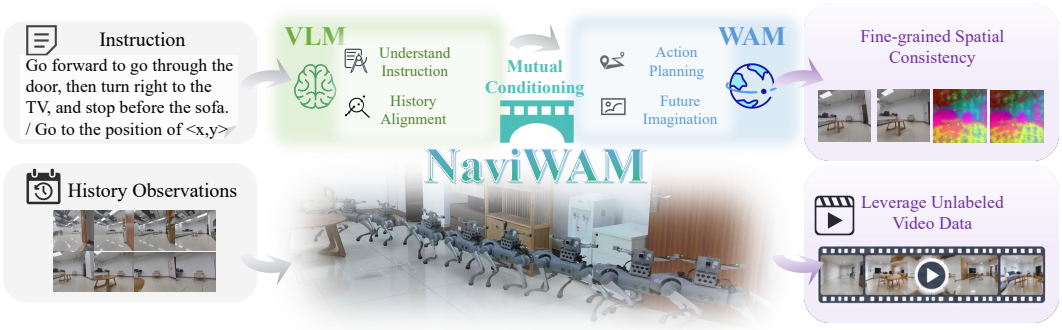

Figure 1: **NaviWAM learns navigation from waypoint labels and video.** A VLM encodes visual history with route instructions or relative-goal text. A shared dual-stream predictor learns waypoint generation and patch-level next-frame prediction. The visual objective uses additional videos during mixed training and is reused for subsequent adaptation without new waypoint supervision. Only waypoints are generated at deployment.

1 INTRODUCTION

Embodied navigation policies must ground task specifications in an evolving visual environment and convert that understanding into executable motion. Large-scale navigation data and pretrained vision–language models (VLMs) have substantially advanced this capability (Wang et al., 2023; Wei et al., 2025). Most trajectory-supervised policies, however, receive a low-dimensional motion target

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

at each decision. This target specifies where to move, but does not directly describe the spatially distributed visual changes observed along the route. Successive images therefore contain useful learning signals beyond those supplied by the trajectory coordinates. A trajectory-only objective also cannot directly exploit videos for which a compatible action representation is unavailable.

Predictive learning offers a way to use this additional information. Recent world–action models combine action prediction with future-state supervision, and extend learning to videos without trajectory annotations (Azuma et al., 2026; Yang et al., 2026; Zhang et al., 2026; Zhou et al., 2026). For long-horizon navigation, the challenge is not simply to predict a future image: the prediction should relate local scene changes to the task and the agent's observation history. The current view alone may not reveal which part of an instruction is relevant, or whether a landmark has already been passed. At the same time, a global image descriptor can discard the local visual structure needed to distinguish different regions of the scene. These considerations motivate a predictive objective that is both task–history conditioned and spatially resolved.

We introduce *NaviWAM*, which couples *trajectory prediction (Policy)* with *next-frame feature prediction (NF)* through shared predictive components. A VLM integrates task specifications with observation history selected by Temporal–Diversity Hybrid Sampling (TDHS). Its hidden states and the current image features condition a dual-stream Multimodal Diffusion Transformer (MMDiT) (Esser et al., 2024). The Policy task predicts a local trajectory; the NF task predicts a grid of visual features extracted from a subsequent frame. Retaining the patch grid supplies localized visual targets without requiring RGB reconstruction.

The two tasks use the same backbone but are invoked separately. Policy calls use a learned future-feature placeholder, so trajectory prediction does not depend on generating a future observation. NF calls use learned action queries rather than recorded trajectories, allowing training from video–instruction pairs without action labels. Within each call, joint attention connects the visual and action streams. Across calls, both losses update the shared predictor and the trainable VLM conditioning layers. Thus, NF supervision can affect the representations used for trajectory prediction while Policy supervision continues to teach the trajectory interface.

We use this design in two stages. Stage I combines labeled navigation sequences, which supervise both Policy and NF, with additional Internet video–instruction pairs (Han et al., 2025; Li et al., 2025) for NF. Stage II reuses NF for offline adaptation to target scenes, together with part of Stage I data replay. Notably, target-scene action labels are not collected; Stage I source replay preserves an explicit action-learning signal throughout adaptation. The adapted model is evaluated on held-out routes in those scenes. At deployment, it samples only the Policy task while retaining both streams of the shared predictor.

Experiments with separately trained VLN and PointGoal models evaluate navigation performance, training objectives, visual representations, history selection, and data composition. Policy + NF improves RxR-CE success rate (SR) from 58.9% to 63.6% over Policy training. Target-scene NF adaptation increases fine-/coarse-grained VLNVerse SR by 6.91/7.06 points, with a separate target-trajectory-supervised reference showing the remaining gap. Our contributions are twofold:

- A task- and history-conditioned navigation framework that couples trajectory prediction with patch-level NF prediction through a shared dual-stream predictor, allowing visual supervision to complement action learning without future visual generation at deployment.
- A two-stage learning scheme that applies the same NF objective to additional video experience and offline target-scene adaptation, supported by navigation benchmarks and analyses of the training objectives, representations, history, and data mixtures.

2 Related Work

2.1 Embodied Navigation

Navigation requires instruction grounding, memory of previous observations, and spatially coherent decisions. Recurrent and history-aware transformers, together with metric or topological representations, address complementary parts of this problem (Hong et al., 2021; Chen et al., 2021; An et al., 2023; Chen et al., 2022; Liu et al., 2026). Large-scale data and VLM-based policies extend the range of visual and linguistic conditions that an agent can handle (Wang et al., 2023; Zhang et al., 2024; Wei et al., 2025), while goal-conditioned policies study transferable local navigation

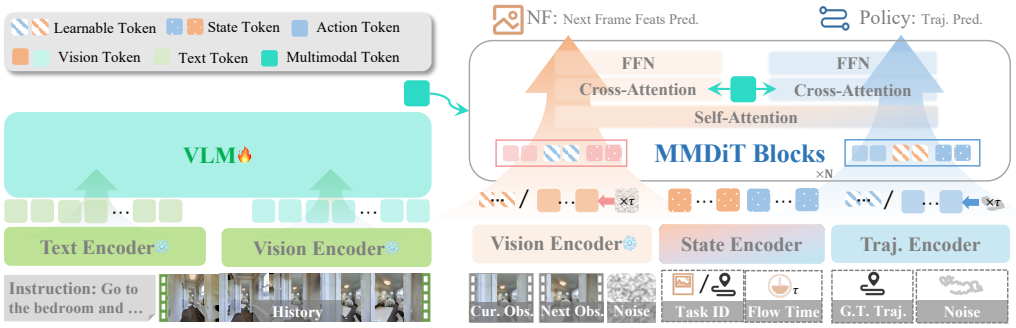

Figure 2: **NaviWAM architecture.** The VLM integrates task text and selected history, and its hidden states condition a shared dual-stream predictor through cross-attention. Joint self-attention connects visual and action tokens. Policy calls use a noisy trajectory and a future-feature placeholder; NF calls use noisy next-frame features and learned action queries. Task and flow-time embeddings select the prediction objective. VLM embeddings and vision encoders are frozen, while language blocks, the multimodal projector, and WAM are trained.

(Wijmans et al., 2020; Cai et al., 2025). NaviWAM builds on task-conditioned trajectory prediction, with an additional NF objective that uses visual changes along a route as direct training targets.

Adaptation addresses a related but distinct question: how to improve a trained agent after receiving experience from a deployment environment. FSTTA studies online adaptation with unlabeled test samples, and NavMorph evolves contextual memory within a predictive navigation model (Gao et al., 2024; Yao et al., 2025). NaviWAM instead performs offline optimization on target-scene video–instruction pairs before evaluation, retaining labeled source replay. Its adapted results measure performance on held-out routes after target-scene exposure, rather than online learning during execution or generalization without such exposure.

2.2 Predictive Learning for Navigation

NavWAM combines future observations, action chunks, and goal-progress values, while WAM-Nav uses asymmetric visual and action horizons (Azuma et al., 2026; Yang et al., 2026). AstraNav-World and WNM-3D also investigate visual–action prediction for VLN (Hu et al., 2025; Huang et al., 2026). FutureNav combines policy learning with forward/inverse dynamics and pooled future spatial features, using the policy alone at inference by default (Zhang et al., 2026). UniNav trains with trajectory-labeled sequences and videos without trajectory labels; its fast variant removes future-video tokens at inference (Zhou et al., 2026). In manipulation, LDA-1B provides the task-conditioned dual-stream latent-prediction design on which we build, and Fast-WAM separates video co-training from test-time future imagination (Lyu et al., 2026; Yuan et al., 2026).

NaviWAM studies this predictive paradigm in long-horizon navigation. Its focus is the combination of task–history conditioning, patch-level NF targets, and a predictor shared with the Policy task. The same NF interface accepts heterogeneous video–instruction pairs during initial training and target-scene adaptation. This positions the contribution in how predictive supervision is used for navigation across the two stages, rather than in dual-stream modeling or rollout-free inference alone.

3 Method

3.1 Overview

At decision step t, the agent receives an RGB observation $\mathbf{I}_t$ and task text $\mathbf{L}$. For vision-and-language navigation (VLN), $\mathbf{L}$ is an unchanging route instruction; for PointGoal, $\mathbf{L}_t$ expresses the real-time relative goal. Given the observation history $[\mathbf{I}_0, \ldots, \mathbf{I}_t]$, we first apply the history sampling strategy to obtain a compact history $\mathbf{O}_{\leq t} = [\mathbf{I}_{t_1}, \ldots, \mathbf{I}_{t_{k_h-1}}, \mathbf{I}_t]$. Conditioned on $\mathbf{O}_{\leq t}$, the agent predicts a local trajectory $\mathbf{A}_t = [\boldsymbol{a}_{t,1}, \ldots, \boldsymbol{a}_{t,k_a}]^\top \in \mathbb{R}^{k_a \times 3}$, where each row contains normalized planar motion and heading components in an egocentric frame. A controller converts the denormalized trajectory into executable motion, and the agent replans as new observations arrive.

As illustrated in Figure 2, NaviWAM couples a navigation policy with a next-frame feature prediction objective within a shared world–action module (WAM). Given the task instruction, long-horizon observation history, current visual observation, and task-specific state input, NaviWAM supports two conditional prediction tasks:

$$\mathbf{A}_t = \mathrm{NaviWAM}_{\mathrm{Policy}}\left(\mathbf{O}_{\le t}, \mathbf{L}, \mathbf{S}_t^{\mathrm{p}}\right), \mathbf{V}_{t+1} = \mathrm{NaviWAM}_{\mathrm{NF}}\left(\mathbf{O}_{\le t}, \mathbf{L}, \mathbf{S}_t^{\mathrm{v}}\right), \tag{1}$$

where the Policy task predicts a local navigation trajectory, while the NF task predicts patch-level visual features of a subsequent frame. The two tasks use task-specific inputs and output heads while sharing the main predictive backbone, allowing next-frame supervision to improve shared representations without requiring trajectory labels. The detailed construction of the visual history, task conditioning, and task-specific inputs is introduced in the following sections.

3.2 VLM-Based Task–History Conditioning

Temporal–Diversity Hybrid Sampling. Long routes contain both recent observations relevant to immediate motion and earlier views useful for instruction grounding. TDHS combines these sources within a budget of up to k_{h} frames, including the current observation. Its temporal component follows the recency-based forgetting strategy of NavFoM (Zhang et al., 2025a). Its diversity component selects visually distinct past observations, reducing redundancy among similar views. For each image, we mean-pool and normalize the VLM vision tokens, apply a fixed-seed projection dimensions d_{h}, and normalize again to obtain a descriptor $\boldsymbol{z}_i$. Diversity selection starts from the current frame, $\mathcal{B}_t = \{t\}$, and repeatedly adds the past frame most distant from its nearest selected descriptor:

$$i^\star = \underset{i \in \{0,\dots,t-1\} \setminus \mathcal{B}_t}{\arg\max} \; \min_{j \in \mathcal{B}_t} (1 - \boldsymbol{z}_i^\top \boldsymbol{z}_j). \tag{2}$$

We merge the temporal and diversity selections, remove duplicates, and arrange the retained observations chronologically to form $\mathbf{O}_{\le t}$. Selection uses only observations available by step t. The compact descriptors determine which frames to retain; the VLM still processes the selected frames through its native visual-token pathway, see Appendix 6.1.2 for details.

Task–history context and trainable modules. Qwen3-VL-4B (Bai et al., 2025) jointly processes the task and selected observations:

$$\mathbf{H}_t = \mathrm{VLM}_{\mathrm{last}}(\mathbf{L}, \mathbf{O}_{\le t}). \tag{3}$$

We use the full final-layer hidden-state sequence as WAM's cross-attention context, rather than requiring an intermediate textual plan. The VLM's *token embeddings and vision encoder are frozen*, whereas its *language blocks and multimodal projector are trainable*. Thus, both Policy and NF losses can refine the task–history representation through the trainable conditioning pathway.

3.3 Coupled World–Action Prediction

WAM uses a dual-stream MMDiT (Esser et al., 2024). Visual and action tokens retain stream-specific transformations while exchanging information through joint self-attention; both streams attend to $\mathbf{H}_t$. A task identifier $q \in \{\mathrm{Policy}, \mathrm{NF}\}$ and flow time $\tau \in [0, 1]$ modulate the blocks through adaptive layer normalization. Task conditioning lets the same backbone operate on two target spaces: local trajectories and next-frame feature grids.

Patch-level visual targets. A frozen encoder supplies the current visual input and the NF target:

$$\mathbf{V}_t = \mathrm{Enc}_{\mathrm{v}}^{\mathrm{wam}}(\mathbf{I}_t), \qquad \mathbf{V}_{t+1}^{\mathrm{gt}} = \mathrm{Enc}_{\mathrm{v}}^{\mathrm{wam}}(\mathbf{I}_{t+1}). \tag{4}$$

We use DINOv2 (Oquab et al., 2024) as $\mathrm{Enc}_{\mathrm{v}}^{\mathrm{wam}}(\cdot)$ by default and retain its $n_{\mathrm{v}} \times d_{\mathrm{v}}$ patch grid. Predicting this grid provides a target for each image region instead of collapsing the frame into one pooled vector. The objective operates in feature space and does not reconstruct RGB pixels. Here, $t + 1$ denotes the next ground-truth frame from the reference path used as a prediction target. Flow time τ is a separate variable governing the generative process.

Flow-matching formulation. Let $\mathbf{Y}_t^{\mathrm{Policy}} = \mathbf{A}_t^{\mathrm{gt}}$ and $\mathbf{Y}_t^{\mathrm{NF}} = \mathbf{V}_{t+1}^{\mathrm{gt}}$. For either task, we interpolate between standard Gaussian noise $\mathbf{N}^q$ and the corresponding target:

$$\mathbf{Y}_{t,\tau}^q = (1 - \tau)\mathbf{N}^q + \tau \mathbf{Y}_t^q. \tag{5}$$

The predictor estimates the velocity $\mathbf{Y}_t^q - \mathbf{N}^q$ along this path. Only the selected target is noised; the current-frame features remain clean in both tasks.

Task-specific inputs. The visual input merges current and next-frame features by concatenating them along the channel dimension before projection. When no future feature is supplied, a learned token is repeated over the patch grid to form $\mathbf{P}^{\mathrm{v}} \in \mathbb{R}^{n_{\mathrm{v}} \times d_{\mathrm{v}}}$. Likewise, learned action queries $\mathbf{P}^{\mathrm{a}}$ replace the trajectory input when no trajectory is supplied. For each prediction task q, a state encoder jointly encodes the task identity and flow time τ into a state representation $\mathbf{S}_t^q$, yielding the two task-specific WAM calls:

$$\begin{aligned} \widehat{\mathbf{F}}_{t,\tau}^{\mathrm{Policy}} &= \mathrm{WAM}_{\mathrm{a}}\Big([\mathbf{V}_t; \mathbf{P}^{\mathrm{v}}], \mathrm{Enc}_{\mathrm{traj}}(\mathbf{Y}_{t,\tau}^{\mathrm{Policy}}, \tau), \mathbf{S}_t^{\mathrm{Policy}}, \mathbf{H}_t\Big), \\ \widehat{\mathbf{F}}_{t,\tau}^{\mathrm{NF}} &= \mathrm{WAM}_{\mathrm{v}}\Big([\mathbf{V}_t; \mathbf{Y}_{t,\tau}^{\mathrm{NF}}], \mathbf{P}^{\mathrm{a}}, \mathbf{S}_t^{\mathrm{NF}}, \mathbf{H}_t\Big). \end{aligned} \tag{6}$$

Here, $\mathrm{WAM}_{\mathrm{a}}(\cdot)$ and $\mathrm{WAM}_{\mathrm{v}}(\cdot)$ select the action and visual output heads of the same backbone. $\mathrm{Enc}_{\mathrm{traj}}(\cdot)$ is the trajectory encoder. A Policy call receives a noisy trajectory and a future-feature placeholder, then supervises only the trajectory velocity. An NF call receives a noisy next-frame feature grid and learned action queries, then supervises only the feature velocity. Consequently, Policy prediction never requires a target or generated future frame, and NF prediction never requires a recorded trajectory.

How NF supervision reaches the policy. Both streams participate in each call. During NF prediction, the learned action queries interact with visual tokens through joint attention, so the NF loss can update the shared action-stream transformations as well as the visual stream. It also reaches the trainable VLM language blocks and projector through $\mathbf{H}_t$. These components are reused in Policy prediction. The trajectory input encoder and output head, in contrast, receive their direct supervision from the Policy loss. This distinction explains why videos can improve shared navigation representations without replacing trajectory supervision. Our use of *coupled* refers to this parameter sharing and cross-stream interaction, rather than joint sampling of physically constrained action–future pairs. Appendix 6.1.1 gives the supervision paths in more detail.

3.4 Training, Adaptation, and Inference

Task objectives. Write $f_q(\mathbf{X}, \tau; \mathbf{H}_t, \mathbf{V}_t)$ for the complete task-q velocity predictor in equation 6. For a task-specific collection dataset $\mathcal{D}_q$, the flow-matching loss is:

$$\mathcal{L}_q(\mathcal{D}_q) = \mathbb{E}_{\mathcal{D}_q, \tau \sim p_\tau, \mathbf{N}^q}\left[\mathrm{mse}\big(f_q(\mathbf{Y}_{t,\tau}^q, \mathbf{S}_t^q, \mathbf{H}_t, \mathbf{V}_t), \mathbf{Y}_t^q - \mathbf{N}^q\big)\right], \tag{7}$$

where MSE averages over target entries and p_τ specifies flow-time sampling.

Stage I: learning from trajectories and videos. Labeled navigation sequences provide both trajectory targets and subsequent frames. Additional video–instruction pairs provide subsequent frames but no Policy targets. The two sources are combined as:

$$\mathcal{L}_{\mathrm{I}} = \mathcal{L}_{\mathrm{Policy}}(\mathcal{D}_{\mathrm{Policy}}^{\mathrm{I}}) + \lambda_1 \mathcal{L}_{\mathrm{NF}}(\mathcal{D}_{\mathrm{NF}}^{\mathrm{I}}), \qquad \lambda_1 > 0, \tag{8}$$

where $\mathcal{D}_{\mathrm{Policy}}^{\mathrm{I}}$ contains labeled navigation samples and $\mathcal{D}_{\mathrm{NF}}^{\mathrm{I}}$ contains their associated observations together with additional video–instruction pairs. Policy supervision anchors the generated trajectory representation, while NF adds a spatially resolved learning signal from both data sources.

Stage II: offline target-scene NF adaptation. After Stage I, we adapt the model using target-scene video–instruction pairs alongside source-navigation and Internet-video replay. Let $\mathcal{D}_{\mathrm{Policy}}^{\mathrm{ret}}$ and $\mathcal{D}_{\mathrm{NF}}^{\mathrm{ret}}$ denote retained source-navigation samples for Policy and NF training, respectively, and let $\mathcal{D}_{\mathrm{NF}}^{\mathrm{web}}$ and $\mathcal{D}_{\mathrm{NF}}^{\mathrm{tgt}}$ denote replayed Internet and target-scene video–instruction pairs for NF. The Stage II objective is

$$\mathcal{L}_{\mathrm{II}} = \mathcal{L}_{\mathrm{Policy}}\left(\mathcal{D}_{\mathrm{Policy}}^{\mathrm{ret}}\right) + \lambda_2 \mathcal{L}_{\mathrm{NF}}\left(\mathcal{D}_{\mathrm{NF}}^{\mathrm{ret}} \cup \mathcal{D}_{\mathrm{NF}}^{\mathrm{web}} \cup \mathcal{D}_{\mathrm{NF}}^{\mathrm{tgt}}\right), \qquad \lambda_2 > 0. \tag{9}$$

Here, $\cup$ denotes sampled mixtures without implying equal source weights. Policy supervision comes exclusively from source replay, while Internet and target-scene videos provide only NF supervision. Policy remains the trajectory-generating task at evaluation.

The shared Policy–NF architecture allows next-frame supervision to update predictive components also used for trajectory prediction. This enables adaptation from target-scene video–instruction pairs without target-scene trajectory labels, while source replay retains Policy supervision. In practice,

this offers a route to adapting navigation models using low-cost human-recorded walkthroughs rather than trajectory-labeled robot demonstrations.

To evaluate this capability under a source-to-target domain shift, we use VLN-CE navigation data in Habitat as the Stage I source and the Isaac scenes used by VLNVerse as the adaptation target. The source scenes are real-world reconstructions with relatively spacious residential layouts, preserving real-world appearance and spatial structure despite reconstruction artifacts. In contrast, the target scenes feature more compact layouts and different lighting and visual appearance (see Figure 3 in Section 4). As a synthetic proxy for human-recorded walkthroughs, we collect target-scene video–instruction pairs along planned trajectories with camera jitter and variable traversal speeds. Collection routes are independent of evaluation routes, and the planned trajectories are withheld from Policy supervision. This defines offline *target-scene adaptation on held-out routes*; Appendix 6.3 details the collection and supervision protocol.

Inference. Sampling begins with $\widehat{\mathbf{Y}}_t^{q,(0)} = \mathbf{N}^q$. Using k_{s} Euler steps and $\tau_i = i/k_{\mathrm{s}}$, we update

$$\widehat{\mathbf{Y}}_t^{q,(i+1)} = \widehat{\mathbf{Y}}_t^{q,(i)} + \frac{1}{k_{\mathrm{s}}} f_q\Big(\widehat{\mathbf{Y}}_t^{q,(i)}, \tau_i; \mathbf{H}_t, \mathbf{V}_t\Big). \tag{10}$$

Navigation samples only $q = \mathrm{Policy}$, then denormalizes the predicted trajectory for execution. The visual stream still processes current features and the learned placeholder, but the NF task is not sampled. This separates using future-frame supervision *during learning* from generating future observations *during control*.

4 Experiments

In this section, we first evaluate navigation performance with separately trained VLN and PointGoal models. We then examine how additional video data, target-scene NF adaptation, the prediction objectives, visual representations, and history selection affect performance.

4.1 Experimental Setup

Training Dataset. For VLN, our policy corpus contains approximately 2.3M processed samples from R2R-CE and RxR-CE, each with an instruction, observations, and a trajectory target. These sequences support both Policy and NF training. Additional NF data come from RoomTour3D (Han et al., 2025), Sekai (Li et al., 2025), and our Isaac video–instruction collection. At the largest mixture, the additional data comprise approximately 0.3M RoomTour3D samples, 0.3M Sekai samples, and the remaining 0.78M Isaac samples, totaling 1.38M. This is 60% of the Policy-data volume, yielding a Policy-to-NF-data ratio of 1:1.6 when policy-associated frames are included. These are processed training samples, not counts of unique raw videos. Additional sequences provide NF targets without trajectory supervision. Appendix 6.3 details the counts and synthetic-data pipeline.

Tasks and benchmarks. VLN evaluation uses R2R-CE and RxR-CE validation-unseen (Krantz et al., 2020; Ku et al., 2020), together with fine- and coarse-grained VLNVerse (Lin et al., 2025). PointGoal follows the InternVLA-N1(S1) protocol on 40 InternScenes environments (Intern Robotics, 2025; Cai et al., 2025; Zhong et al., 2025). VLNVerse-PG replaces instructions in VLNVerse-Fine-Unseen with relative goals; VLNVerse-PG-NPC adds moving NPC obstacles. For PointGoal, RGB-only describes the *visual* input: the policy also receives the task-provided relative goal. We report success rate (SR) and success weighted by path length (SPL), as well as navigation error (NE), oracle success (OS), and normalized dynamic time warping (nDTW) where available (Anderson et al., 2018; Ilharco et al., 2019). NE is measured in meters; other metrics are percentages. The target-scene adaptation study uses the VLN model.

Implementation details. NaviWAM uses Qwen3-VL-4B and a randomly initialized 256.6M-parameter WAM. The reported training run uses 32 H100 GPUs for approximately one day. Token embeddings and both visual encoders are frozen; the VLM language blocks, multimodal projector, and WAM are trainable. Detailed training and inference hyperparameters are provided in Appendix 6.2. Benchmark tables compare the specified configurations using their reported observation modalities. Real-world examples use remote RTX 4090 inference and onboard trajectory execution; they are qualitative demonstrations.

Table 1: **Comparison on continuous VLN benchmarks.** Performance on the unseen splits of R2R-CE and RxR-CE. In *Obs.*, Depth and Odom. indicate the use of depth and odometry during training and/or evaluation. Bold and underlined values denote the best and second-best results. For FutureNav, "**0k**" denotes training on VLN-CE expert trajectories without additional navigation data.

Method	Obs.		R2R Val-Unseen				RxR Val-Unseen			
	Depth	Odom.	NE↓	OS↑	SR↑	SPL↑	NE↓	nDTW↑	SR↑	SPL↑
Multi-view methods										
CMA* (Krantz et al., 2020)	✓	✓	6.20	52.0	41.0	36.0	8.76	–	26.5	22.1
ETPNav* (An et al., 2024)	✓	✓	**4.71**	65.0	57.0	49.0	5.64	–	54.7	44.8
InstructNav (Long et al., 2024)	✓	✓	6.89	–	31.0	24.0	–	–	–	–
Monocular methods										
NaVid (Zhang et al., 2024)	–	–	5.47	49.1	37.4	35.9	–	–	–	–
Uni-NaVid (Zhang et al., 2025b)	–	–	5.58	53.3	47.0	42.7	6.24	–	48.7	40.9
NaVILA (Cheng et al., 2025)	–	–	5.22	62.5	54.0	49.0	6.77	–	49.3	44.0
StreamVLN (Wei et al., 2025)	–	–	<u>4.98</u>	64.2	56.9	51.9	6.22	–	52.9	46.0
InternVLA-N1 (InternVLA-N1 Team, 2025)	✓	–	**4.83**	63.3	<u>58.2</u>	<u>54.0</u>	5.91	**65.3**	53.5	46.1
Qwen-VLA (Qwen Team, 2026)	–	–	5.10	**69.0**	57.5	51.2	5.80	–	<u>59.6</u>	47.8
NavFoM (Zhang et al., 2025a)	–	–	5.01	64.9	56.2	51.2	<u>5.51</u>	60.2	57.4	<u>49.4</u>
FutureNav-4B (**0k**) (Zhang et al., 2026)	–	–	5.13	61.8	55.1	50.1	5.62	61.8	54.5	46.0
FutureNav-8B (**0k**) (Zhang et al., 2026)	–	–	5.15	61.6	55.5	51.4	5.93	59.8	53.3	45.4
NaviWAM	–	–	5.01	<u>67.4</u>	**59.8**	**54.7**	**5.35**	<u>63.1</u>	**63.6**	**52.5**

Table 2: **Comparison on point-goal navigation.** Performance on Sys1-PG Home, Sys1-PG Commercial, VLNVerse-PG, and VLNVerse-PG-NPC. VLNVerse-PG converts the instructions in VLNVerse-Fine-Unseen into point-goal inputs, while VLNVerse-PG-NPC further introduces moving NPCs as dynamic obstacles along the navigation routes. In *Obs.*, S.RGB and Depth indicate the use of front-facing RGB observations and depth information, respectively.

Method	Obs.		Sys1-PG Home		Sys1-PG Commercial		Vlnverse-PG		Vlnverse-PG-NPC	
	S.RGB	Depth	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
DD-PPO Wijmans et al. (2020)	✓	✓	0.4	0.4	5.3	5.2	–	–	–	–
LoGoPlanner Peng et al. (2025)	✓	✓	57.3	52.4	67.1	63.9	–	–	–	–
InternVLA-N1(S1) Intern Robotics (2025)	✓	✓	60.0	<u>55.6</u>	71.4	68.2	–	–	–	–
NavDP Cai et al. (2025)	✓	✓	<u>60.3</u>	54.7	<u>74.1</u>	70.5	83.6	83.2	71.9	71.6
NaviWAM	✓	–	**77.4**	**70.2**	**74.7**	**71.7**	**87.0**	**83.7**	**81.1**	**78.5**

4.2 Benchmark Results

Vision-and-language navigation. As shown in Table 1, NaviWAM achieves 59.8% SR and 54.7% SPL on R2R-CE, and 63.6% SR and 52.5% SPL on RxR-CE. Compared with the best baseline for each metric, these represent gains of 1.6/0.7 and 4.0/3.1 percentage points in SR/SPL, respectively. These results demonstrate improved task completion with RGB-only observations.

Point-goal navigation. In Table 2, NaviWAM improves over the best baselines by 17.1/14.6 percentage points in SR/SPL on Sys1-PG Home and 0.60/1.15 points on Sys1-PG Commercial, showing larger gains in residential scenes despite using no depth input. On VLNVerse-PG, the SR/SPL margins increase from 3.42/0.47 points in the static setting to 9.2/6.9 points with moving NPCs. The static-to-dynamic SR drop is also smaller (5.9 versus 11.7 points), suggesting greater robustness to the evaluated dynamic obstacles.

VLNVerse and additional action supervision. Table 3 shows that the base model surpasses the best baseline on fine-grained navigation by 11.5/5.8 percentage points in SR/SPL, but remains less competitive on coarse-grained instructions. Additional action-labeled VLNVerse training yields the best results on both splits: 79.5% SR and 75.8% SPL on fine-grained navigation, and 63.4% SR and 59.0% SPL on coarse-grained navigation. The corresponding SR gains over the base model are 19.4 and 28.3 points, highlighting the benefit of in-domain action-labeled training, particularly for coarse-grained navigation. Evaluation details of VLNVerse-PG and VLNVerse-PG-NPC are provided in the Appendix.

4.3 Additional NF Data and Target-Scene Adaptation

Scaling additional NF data. Figure 3(a) fixes the Policy corpus and increases the amount of NF data. The ratio 1:1 includes only observations associated with the 2.3M Policy samples; ratios 1:1.2, 1:1.4, and 1:1.6 introduce an additional 0.46M, 0.92M, and 1.38M NF samples, respectively. Across

Method	Obs.	F-G		C-G	
		SR↑	SPL↑	SR↑	SPL↑
CMA (Krantz et al.)	Pano.	31.2	27.9	35.5	32.5
RDP (Wang et al.)	Pano.	48.6	42.7	40.9	35.1
Seq2Seq (Sutskever et al.)	Pano.	35.0	33.4	33.1	31.0
HNR (Wang et al.)	Pano.	33.0	29.6	40.9	37.3
NaviWAM	Single	60.1	48.5	35.1	30.3
NaviWAM (w/ action)	Single	**79.5**	**75.8**	**63.4**	**59.0**

Table 3: **VLNVerse Results.** Performance on val-unseen splits of the fine- and coarse-grained tasks.

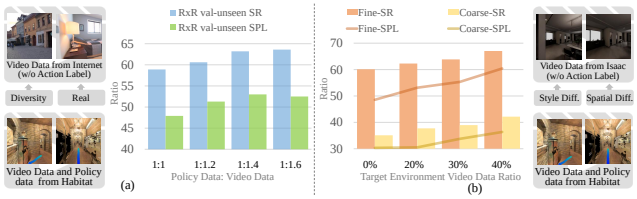

Figure 3: **Effect of training data composition.** (a) Effect of the video-to-policy data ratio on RxR-CE Unseen. (b) Effect of the proportion of target-domain (Isaac) data in Stage II on VLNVerse-Fine Unseen.

Table 4: Ablation on Vision Encoders in WAM.

Encoder	NE↓	nDTW↑	SR↑	SPL↑
SigLIP2	6.08	60.8	59.7	50.0
Qwen3-VL	5.72	60.5	61.0	50.6
DINOv2	5.35	63.1	63.6	52.5

Table 5: Ablation on Training Objectives in WAM.

Task	xSR↑	xSPL↑	fSR↑	fSPL↑
Policy	58.9	47.9	57.6	47.0
Policy + NF + FD	64.9	54.6	54.5	46.6
Policy + NF	63.6	52.5	60.1	48.5

Table 6: Ablation on History Sampling Strategies in VLM.

Sampling	NE↓	nDTW↑	SR↑	SPL↑
Diversity Only	5.77	60.3	56.9	46.9
Temporal Only	5.72	61.0	60.6	51.3
TDHS	5.35	63.1	63.6	52.5

these augmented settings, RxR-CE SR increases from 60.6 to 63.2 and 63.6, while SPL rises from 51.3 to 53.0 before slightly declining to 52.5. These results suggest diminishing returns from scaling NF data: the largest mixture yields only a marginal SR gain and no further improvement in SPL. A plausible explanation is that additional videos broaden visual coverage but may provide increasingly redundant or less navigation-relevant supervision, limiting their marginal contribution to trajectory prediction. Effective scaling may therefore depend on the relevance and diversity of the added data, rather than volume alone.

Target-scene NF adaptation. Figure 3(b) reports target-scene video proportions of 0%, 20%, 30%, and 40%. At fixed total video volume, target-scene Isaac sequences replace part of the Internet-video mixture. Policy supervision continues on source trajectories, while both types of video supply NF supervision. All four metrics increase across the reported settings. At the final setting, fine-grained SR/SPL improve from 60.12/48.54 to 67.03/60.42, and coarse-grained SR/SPL improve from 35.10/30.32 to 42.16/36.36. The corresponding gains are 6.91/11.88 and 7.06/6.04 points.

4.4 Ablation Study

Prediction objectives. Policy + NF improves SR/SPL over Policy by 4.7/4.6 points on RxR-CE and 2.5/1.5 on VLNVerse-Fine (Table 5). This comparison evaluates NF together with its associated data. Adding forward dynamics (FD), which conditions next-frame feature prediction on the ground-truth trajectory, further improves RxR-CE by 1.3/2.1 points but reduces VLNVerse-Fine by 5.6/1.9. Policy + NF is our default: it improves over Policy on both benchmarks without the cross-benchmark degradation observed when FD is added. Unlike NF, FD requires a trajectory condition and therefore cannot use every video–instruction pair through the same interface. These results suggest a trade-off: FD may better capture environment-specific information, but its benefits do not consistently translate into improved cross-domain generalization.

Visual representations. We replace WAM's visual encoder while preserving the VLM task–history interface in Table 4. Each replacement changes both the current-frame input and the NF target. DINOv2 (Oquab et al., 2024) outperforms SigLIP2 (Tschannen et al., 2025) and Qwen3-VL (Bai et al., 2025), improving SR/SPL over Qwen3-VL by 2.6/1.9 points. Its advantage also appears in NE and nDTW, supporting DINOv2 as the default representation for the shared predictor. Because inputs and targets change together, this is an encoder-configuration comparison rather than an isolated test of the NF target.

History selection. In Table 6, TDHS achieves 63.6 SR and 52.5 SPL, compared with 60.6/51.3 for temporal-only sampling and 56.9/46.9 for diversity-only sampling. Retaining visual diversity in addition to recent frames improves SR by 3.0 points, while adding temporal context to diversity sampling improves it by 6.7. The combined sampler also gives the lowest NE and highest nDTW

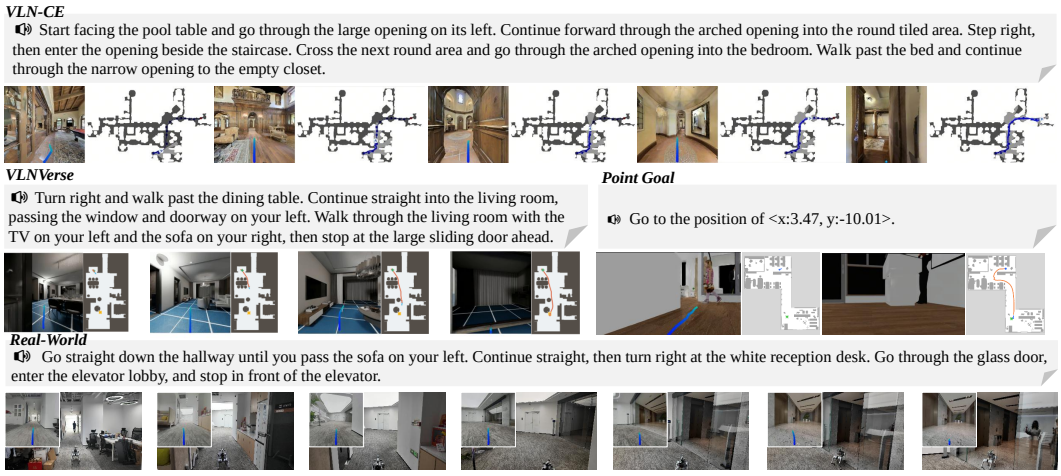

Figure 4: **Qualitative examples across environments and benchmarks.** Representative visualizations on VLN-CE, VLNVerse, point-goal navigation, and real-world environments.

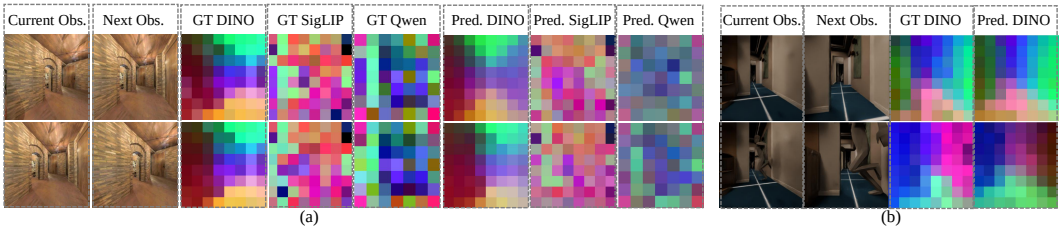

Figure 5: **Visualization of visual representations and future feature prediction.** (a) Two consecutive observations and their representations encoded by DINOv2, SigLIP2, and Qwen3-VL, along with the future representations predicted by NaviWAM. (b) NaviWAM captures the approximate shape of unseen dynamic obstacles in the predicted DINOv2 feature space, despite being trained without dynamic obstacles.

among these variants. The results support using recency and visual diversity together to select the route history presented to the VLM.

4.5 Qualitative Examples and Discussion

Figure 4 illustrates navigation under different task specifications in simulation and real-world scenes. The real-world examples demonstrate the same task-to-trajectory interface with remote inference and onboard control. They complement the quantitative simulation results without constituting a separate real-world benchmark. Figure 5 visualizes NF feature grids for different encoders and scenes with moving NPCs. The examples make the spatial organization of the prediction targets visible and complement the encoder ablation. Their colors are visualization outputs, not calibrated labels of traversability or prediction accuracy.

5 Conclusion

We presented NaviWAM, which couples trajectory prediction (Policy) with patch-level next-frame prediction (NF) under task–history conditioning. A shared dual-stream predictor and trainable VLM conditioning layers allow NF supervision to complement action learning from labeled trajectories and additional video–instruction pairs. The same objective supports target-scene adaptation with source Policy replay, while deployment samples only the Policy task. Experiments demonstrate the effectiveness of this combined scheme and identify remaining gaps in coarse-grained navigation and performance relative to target-trajectory supervision. Further work will isolate the contribution of shared prediction and strengthen grounding under coarse instructions.

AI USE STATEMENT

Generative AI tools were used to assist with English translation and language editing, manuscript organization and drafting, LaTeX formatting, and the generation of visual references for figure design. They were not used to design or conduct experiments or to generate experimental data. The authors reviewed and verified all AI-assisted content and assume full responsibility for the final content of this work.

REFERENCES

Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbert: Multimodal map pre-training for language-guided navigation. In *ICCV*, 2023. URL https://arxiv.org/abs/2212.04385.

Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2024.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3674–3683, 2018.

Daichi Azuma, Taiki Miyanishi, Koya Sakamoto, Shuhei Kurita, Yaonan Zhu, Petr Khrapchenkov, Motoaki Kawanabe, Yusuke Iwasawa, and Yutaka Matsuo. NavWAM: A navigation world action model for goal-conditioned visual navigation. *arXiv preprint arXiv:2606.13494*, 2026. URL https://arxiv.org/abs/2606.13494.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Wenzhe Cai, Jiaqi Peng, Yuqiang Yang, Yujian Zhang, Meng Wei, Hanqing Wang, Yilun Chen, Tai Wang, and Jiangmiao Pang. Navdp: Learning sim-to-real navigation diffusion policy with privileged information guidance. *arXiv preprint arXiv:2505.08712*, 2025.

Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *NeurIPS*, 2021. URL https://arxiv.org/abs/2110.13309.

Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *CVPR*, 2022. URL https://arxiv.org/abs/2202.11742.

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Xueyan Zou, Jan Kautz, Erdem Biyik, Hongxu Yin, Sifei Liu, and Xiaolong Wang. NaVILA: Legged robot vision-language-action model for navigation. In *RSS*, 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Junyu Gao, Xuan Yao, and Changsheng Xu. Fast-slow test-time adaptation for online vision-and-language navigation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 14902–14919. PMLR, 2024.

Mingfei Han, Liang Ma, Kamila Zhumakhanova, Ekaterina Radionova, Jingyi Zhang, Xiaojun Chang, Xiaodan Liang, and Ivan Laptev. Roomtour3d: Geometry-aware video-instruction tuning for embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27586–27596, 2025.

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln bert: A recurrent vision-and-language bert for navigation. In *CVPR*, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Hong_VLN_BERT_A_Recurrent_Vision-and-Language_BERT_for_Navigation_CVPR_2021_paper.html.

Junjun Hu, Jintao Chen, Haochen Bai, Minghua Luo, Shichao Xie, Ziyi Chen, Fei Liu, Zedong Chu, Xinda Xue, Botao Ren, Xiaolong Wu, Mu Xu, and Shanghang Zhang. Astranav-world: World model for foresight control and consistency, 2025.

Yuehao Huang, Yunzi Wu, Xiaotao Zhang, Xinhai Li, Jiankun Dong, Jiajun Lv, Chi Zhang, Chenjia Bai, Yong Liu, and Xuelong Li. WNM-3D: A world navigation model with 3D scene conditioning for closed-loop VLN. *arXiv preprint arXiv:2608.07267*, 2026. URL https://arxiv.org/abs/2608.07267.

Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. In *NeurIPS Workshop on Visually Grounded Interaction and Language*, 2019.

Intern Robotics. Internvla-n1: An open dual-system vision-language navigation foundation model with learned latent plans, 2025.

InternVLA-N1 Team. Internvla-n1: An open dual-system vision-language navigation foundation model with learned latent plans. Technical report, 2025. URL https://internrobotics.github.io/internvla-n1.github.io/.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pp. 104–120. Springer, 2020.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4392–4412, 2020.

Zhen Li, Chuanhao Li, Xiaofeng Mao, Shaoheng Lin, Ming Li, Shitian Zhao, Zhaopan Xu, Xinyue Li, Yukang Feng, Jianwen Sun, Zizhen Li, Fanrui Zhang, Jiaxin Ai, Zhixiang Wang, Yuwei Wu, Tong He, Yunde Jia, and Kaipeng Zhang. Sekai: A video dataset towards world exploration. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Sihao Lin, Zerui Li, Xunyi Zhao, Gengze Zhou, Liuyi Wang, Rong Wei, Rui Tang, Juncheng Li, Hanqing Wang, Jiangmiao Pang, Anton van den Hengel, Jiajun Liu, and Qi Wu. Vlnverse: A benchmark for vision-language navigation with versatile, embodied, realistic simulation and evaluation. *arXiv preprint arXiv:2512.19021*, 2025.

Jiahang Liu, Tianyu Xu, Jiawei Chen, Lu Yue, Jiazhao Zhang, Zhiyong Wang, Minghan Li, Qisheng Zhao, Anqi Li, Qi Su, Zhizheng Zhang, and He Wang. Span-nav: Generalized spatial awareness for versatile vision-language navigation. *arXiv preprint arXiv:2603.09163*, 2026.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. *arXiv preprint arXiv:2406.04882*, 2024.

Jiangran Lyu, Kai Liu, Xuheng Zhang, Haoran Liao, Yusen Feng, Wenxuan Zhu, Tingrui Shen, Jiayi Chen, Jiazhao Zhang, Yifei Dong, Wenbo Cui, Senmao Qi, Shuo Wang, Yixin Zheng, Mi Yan, Xuesong Shi, Haoran Li, Dongbin Zhao, Ming-Yu Liu, Zhizheng Zhang, Li Yi, Yizhou Wang, and He Wang. LDA-1B: Scaling latent dynamics action model via universal embodied data ingestion. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.210. URL https://www.roboticsproceedings.org/rss22/p210.html.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Jiaqi Peng, Wenzhe Cai, Yuqiang Yang, Tai Wang, Yuan Shen, and Jiangmiao Pang. Logoplanner: Localization grounded navigation policy with metric-aware visual geometry. *arXiv preprint arXiv:2512.19629*, 2025.

Qwen Team. Qwen-VLA: Unifying vision-language-action modeling across tasks, environments, and robot embodiments. *arXiv preprint arXiv:2605.30280*, 2026.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Liuyi Wang, Xinyuan Xia, Hui Zhao, Hanqing Wang, Tai Wang, Yilun Chen, Chengju Liu, Qijun Chen, and Jiangmiao Pang. Rethinking the embodied gap in vision-and-language navigation: A holistic study of physical and visual disparities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9455–9465, 2025.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, Junjie Hu, Ming Jiang, and Shuqiang Jiang. Lookahead exploration with neural radiance representation for continuous vision-language navigation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. Scaling data generation in vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12009–12020, 2023. URL https://arxiv.org/abs/2307.15644.

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, et al. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025.

Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. DD-PPO: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *International Conference on Learning Representations (ICLR)*, 2020.

Ning Yang, Yan Huang, Kaiwen Peng, Ziheng He, Kai Wang, Cui Miao, Kailin Lyu, Guo Li, Xiaofeng Wang, Zheng Zhu, Jing Liu, and Nianfeng Liu. Wam-nav: Asymmetric latent world-action modeling for unified visual navigation. *arXiv preprint arXiv:2606.04907*, 2026. URL https://arxiv.org/abs/2606.04907.

Xuan Yao, Junyu Gao, and Changsheng Xu. Navmorph: A self-evolving world model for vision-and-language navigation in continuous environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5536–5546, 2025.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026. URL https://arxiv.org/abs/2603.16666.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. NaVid: Video-based VLM plans the next step for vision-and-language navigation. *Robotics: Science and Systems*, 2024.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, et al. Embodied navigation foundation model. *arXiv preprint arXiv:2509.12129*, 2025a.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-NaVid: A video-based vision-language-action model for unifying embodied navigation tasks. *Robotics: Science and Systems*, 2025b.

Lingfeng Zhang, Zeying Gong, Xiaoshuai Hao, Haoxiang Fu, Qiang Zhang, Mingliang Zhou, Hangjun Ye, Xiaojun Liang, Junwei Liang, and Wenbo Ding. FutureNav: Unified world-action modeling for vision-and-language navigation. *arXiv preprint arXiv:2606.30367*, 2026. URL https://arxiv.org/abs/2606.30367.

Weipeng Zhong, Peizhou Cao, Yichen Jin, Luo Li, Wenzhe Cai, Jingli Lin, Hanqing Wang, Zhaoyang Lyu, Tai Wang, Xudong Xu, Bo Dai, and Jiangmiao Pang. Internscenes: A large-scale simulatable indoor scene dataset with realistic layouts. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Changqing Zhou, Yueru Luo, Zeyu Jiang, and Changhao Chen. UniNav: A unified world-action diffusion model for visual navigation. *arXiv preprint arXiv:2608.03244*, 2026. URL https://arxiv.org/abs/2608.03244.

6 APPENDIX

6.1 ADDITIONAL METHOD DETAILS

6.1.1 SHARED PREDICTION AND SUPERVISION PATHS

NaviWAM uses a shared dual-stream predictor for trajectory prediction (Policy) and next-frame feature prediction (NF). In a Policy call, the trajectory stream receives noisy waypoints, while the visual stream receives current-frame features and a learned future-feature placeholder. In an NF call, the visual stream receives current-frame features and a noisy future-feature grid, while learned action queries replace the trajectory input. Task conditioning selects the prediction objective, and each call supervises its corresponding velocity field.

Shared and task-specific parameters. The visual input merger and dual-stream backbone are reused by both tasks. During NF prediction, learned action queries interact with visual tokens through joint self-attention, providing a gradient path from the NF loss to the shared visual- and action-stream transformations. The loss also reaches the trainable VLM language blocks and multimodal projector through the task–history context. These components are subsequently reused for Policy prediction. The trajectory input encoder and trajectory output head receive supervision from the Policy loss. Source-navigation replay therefore retains a direct training signal for the trajectory interface during video-based adaptation.

The forward-dynamics (FD) variant in the objective ablation conditions future-feature prediction on the ground-truth trajectory. NF instead uses learned action queries, allowing the same prediction interface to process video–instruction pairs without waypoint labels.

Patch-level prediction targets. NF predicts the spatial grid of features extracted from the subsequent frame by the frozen visual encoder. Each target token corresponds to an image patch, preserving local visual information rather than reducing the frame to a single pooled descriptor. The prediction objective operates directly in feature space, without RGB reconstruction. The current-frame features remain clean conditioning inputs in both tasks.

6.1.2 HISTORY SELECTION AND FEATURE REUSE



Figure 6: Examples of historical frames sampled using a hybrid strategy of temporal and similar-frame sampling.

TDHS combines recency-based sampling with visual-diversity selection, as illustrated in Figure 6. For each observation, we spatially mean-pool the VLM vision features, normalize the resulting vector, project it to $d_{\mathrm{h}} = 256$ dimensions using a fixed random seed, and normalize it again. Cosine distance between these descriptors guides farthest-point sampling, initialized with the current frame.

Table 7: Model, history-selection, and flow-matching settings. The history-selection parameters apply to the base VLN configuration; PointGoal uses the current frame only.

Parameter	Value
VLM initialization	Qwen3-VL-4B-Instruct
WAM visual encoder	DINOv2-S
WAM parameter count	256.6M
Descriptor dimension d_{h}	256
VLN input-frame budget	32, including current frame
PointGoal input-frame budget	1, current frame only
Visual share of non-mandatory slots	0.5
Minimum visual / temporal quotas	4 / 4
Descriptor projection seed	3407
Similarity tie tolerance	0.001
Number of flow-time buckets	1000
Training noise repetitions per sample	4
Inference solver	Explicit Euler
Inference integration steps	8

The diversity component retains distinct past views, while the temporal component preserves recent context. We merge the selected indices, remove duplicates, and sort the retained frames chronologically. Frame selection uses only observations available at the current decision step. The compact descriptors are used for selection; the selected frames enter the VLM through its native visual-token pathway.

Frozen per-frame visual features and selection descriptors can be cached for reuse. The VLM's multimodal context is computed from the task text and the selected observation sequence.

6.2 Implementation Details and Hyperparameters

Model initialization and trainable modules. We initialize the VLM with Qwen3-VL-4B-Instruct and use pretrained DINOv2-S as the WAM visual encoder. The 256.6M-parameter WAM, including its MMDiT backbone and trajectory encoder and decoder, is trained from scratch. We train the VLM language blocks, multimodal projector, and WAM. The VLM token embeddings, visual backbone, language-model output head, and DINOv2-S remain frozen. VLN and PointGoal use separately trained models.

Task-specific visual inputs. The VLN configuration uses up to 32 input frames, including the current observation. TDHS, implemented as TV-mix, combines anchor frames, visual-diversity selection, and recency-based forgetting-curve sampling. The visual-selection ratio applies to the non-mandatory frame budget; the visual and temporal quotas are allocated within the total budget. The PointGoal model uses only the current RGB observation, $\mathbf{O}_{\le t} = [\mathbf{I}_t]$, together with the task-provided relative goal expressed as text. Subsequent frames supply NF targets during training. Table 7 summarizes the model, sampling, and inference settings.

Flow matching and inference. For $q \in \{\mathrm{Policy}, \mathrm{NF}\}$, the targets are $\mathbf{Y}_t^{\mathrm{Policy}} = \mathbf{A}_t^{\mathrm{gt}}$ and $\mathbf{Y}_t^{\mathrm{NF}} = \mathbf{V}_{t+1}^{\mathrm{gt}}$. Following equation 5, the corrupted input and velocity target are

$$\mathbf{Y}_{t,\tau}^{q} = (1-\tau)\mathbf{N}^{q} + \tau\mathbf{Y}_t^{q}, \qquad \mathbf{F}_t^{q,\star} = \mathbf{Y}_t^{q} - \mathbf{N}^{q}, \tag{11}$$

where $\mathbf{N}^q$ is standard Gaussian noise with the same shape as the target. Training uses 1000 flow-time buckets and four noise realizations per sample. Navigation step t and flow time τ are distinct: $t+1$ denotes the subsequent observation selected as the NF target, whereas τ indexes the noise-to-target interpolation.

At inference, we use eight explicit Euler steps with $\tau_i = i/8$ and $\Delta\tau = 1/8$ for $i = 0, \ldots, 7$, following equation 10. Navigation invokes only Policy, denormalizes the predicted trajectory, and passes it to the controller for execution. The visual stream processes the current features and learned placeholder without sampling an NF prediction.

Adaptive loss weighting. We balance Policy and NF supervision using an adaptive coefficient computed from running loss statistics. Let j index optimizer updates. The training objective is

$$\mathcal{L}_j = \mathcal{L}_{\text{Policy}} + \lambda_j \mathcal{L}_{\text{NF}}, \qquad \lambda_j = 0.35\, s_j. \tag{12}$$

Both stages use this weighting scheme: λ_1 and λ_2 in the main-text objectives are instantiated by the update-dependent coefficient λ_j in their respective stages. The Policy coefficient remains fixed at 1. We set $s_j = 1$ for the first 1,500 optimizer updates. Subsequently, the scale is adjusted using exponential moving averages (EMAs) of the Policy loss and the base-weighted NF loss, with an EMA decay of 0.98 and a target NF-to-Policy loss contribution ratio of 0.7. The Policy loss EMA is lower-bounded by 0.04 when computing the scale. We clip s_j to $[0.05, 1]$, giving an effective NF weight in $[0.0175, 0.35]$. The statistics are detached from the computation graph and maintained independently on each distributed training worker.

6.3 Video Data and Supervision Accounting

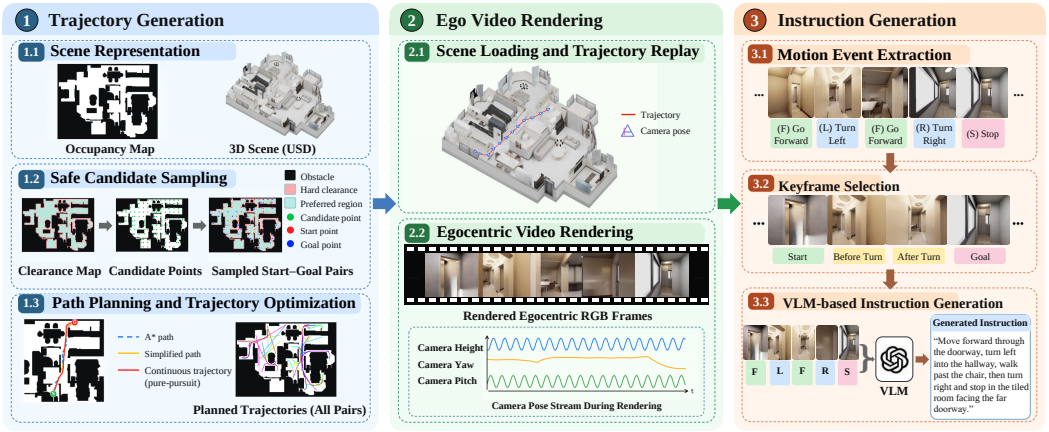

Figure 7: Automated generation of egocentric navigation videos and instructions. We first sample start–goal pairs with sufficient obstacle clearance from scene occupancy maps and generate continuous navigation trajectories through path planning and trajectory refinement. The trajectories are then replayed in 3D scenes to render egocentric RGB videos with variations in camera height and orientation. Finally, we extract motion events, including forward movement, turns, and stopping, and select keyframes at the start, before and after turns, and at the goal. A VLM combines these visual observations with the motion sequence to generate corresponding navigation instructions, yielding paired video–instruction data.

6.3.1 Data Pools and Sample Counts

The source VLN corpus contains approximately 2.3M processed samples from R2R-CE and RxR-CE. Each sample includes a task instruction, visual observations, and corresponding waypoint labels. Its visual sequence supplies an NF target, while its waypoint labels supply a Policy target. Additional video–instruction pairs are sampled from RoomTour3D, Sekai, and the Isaac collection and are used only for NF supervision.

Table 8 lists the available pools. Counts refer to processed training samples or clips, rather than unique raw videos or scenes. The selected subset from each pool depends on the training mixture.

6.3.2 Additional-Video Sampling and Policy-Update Matching

Policy-to-NF sample ratio. Let $N_{\text{p}} = 2.3\text{M}$ denote the number of source Policy samples, and let N_{v} denote the number of NF samples, including those derived from the source VLN corpus. A Policy-to-NF ratio of $1{:}r$ specifies

$$N_{\text{v}} = rN_{\text{p}}, \qquad N_{\text{extra}} = (r-1)N_{\text{p}}. \tag{13}$$

At 1:1, both Policy and NF are trained using the same 2.3M source VLN samples, with no additional videos. Increasing the ratio adds video–instruction samples for NF while leaving the source Policy

Table 8: Available training pools. Counts are approximate. Source navigation supplies both Policy and NF targets; the additional video pools supply NF targets only.

Pool	Size	Counted unit	Supervision
R2R-CE + RxR-CE	2.3M	Navigation samples	Policy + NF
RoomTour3D	0.3M	Constructed clips	NF
Sekai	0.3M	Sampled clips	NF
Isaac collection	1.0M	Video–instruction samples	NF

Table 9: Video-amount study on RxR-CE validation-unseen. Every configuration trains both Policy and NF. The source Policy corpus and number of Policy-supervised optimizer updates are matched across configurations. Sample counts describe the NF corpus; SR and SPL are percentages.

Policy : NF samples	Total NF samples	Additional NF samples	SR	SPL
1:1	2.30M	0	58.9	47.9
1:1.2	2.76M	0.46M	60.6	51.3
1:1.4	3.22M	0.92M	63.2	53.0
1:1.6	3.68M	1.38M	63.6	52.5

corpus unchanged. The 1:1.2, 1:1.4, and 1:1.6 configurations therefore add 0.46M, 0.92M, and 1.38M NF samples, respectively.

For each configuration, the additional samples are drawn from the RoomTour3D, Sekai, and Isaac pools. We record the source-wise composition of the selected samples, including the contribution from Isaac scenes. At 1:1.6, the selected additional data comprise approximately 0.3M RoomTour3D samples, 0.3M Sekai samples, and 0.78M Isaac samples. Together with the source-associated NF samples, this yields 3.68M NF samples. The 1.0M Isaac count in Table 8 denotes the available pool, from which the 0.78M samples are selected for this configuration.

Matched Policy optimization. In the data-amount study, each configuration traverses its NF corpus for one epoch. With more NF samples, the mixed-training schedule also produces more source-Policy updates. We therefore match the number of Policy-supervised optimizer updates across all four configurations to that of the largest-video configuration, 1:1.6. Configurations with fewer videos receive additional Policy updates sampled from the same source VLN corpus until this common budget is reached.

Thus, both the waypoint-labeled dataset and the number of Policy-supervised optimizer updates are fixed across the data-amount comparison. Repeated sampling for these additional updates does not change the dataset counts in Table 9. The varying quantity is the amount of additional NF video data.

Table 9 gives the scores for Figure 3(a). Among the configurations with additional videos, SR increases from 60.6 to 63.2 and 63.6, while SPL reaches 53.0 at 1:1.4 and is 52.5 at 1:1.6. The largest mixture achieves the highest SR in this sweep, whereas the intermediate mixture achieves the highest SPL.

6.3.3 Synthetic Video–Instruction Construction

Figure 7 illustrates the Isaac data-collection pipeline. Occupancy and clearance maps identify feasible start–goal pairs. Planned paths are refined into continuous trajectories and replayed in the simulator to render egocentric RGB sequences. Camera pose and traversal speed are varied during rendering.

We extract motion events along each route and select keyframes at the start, around turns, and at the goal. A VLM uses the selected visual observations and motion events to generate the corresponding route instruction. The resulting video–instruction pairs supply NF training samples. Scene geometry and planned trajectories are used to construct the recordings; the generated trajectories are not used as Policy targets in the video-based training recipe.

Table 10: Target-scene video composition study on VLNVerse. The proportions correspond to Figure 3(b). SR and SPL are percentages.

	VLNVerse-Fine		VLNVerse-Coarse	
Target-scene video proportion	SR	SPL	SR	SPL
0%	60.12	48.54	35.10	30.32
20%	62.30	53.12	37.72	30.52
30%	63.85	55.32	38.96	33.74
40%	67.03	60.42	42.16	36.36

Table 11: FutureNav configurations and additional-data context. FutureNav's external-data counts follow its original reporting convention. NaviWAM uses 1.38M additional NF-only samples. SR and SPL are percentages.

		R2R-CE		RxR-CE	
Configuration	Additional data	SR	SPL	SR	SPL
FutureNav-4B, VLN-CE only	—	55.1	50.1	54.5	46.0
FutureNav-8B, VLN-CE only	—	55.5	51.4	53.3	45.4
FutureNav-4B, full data	10,457K (reported)	65.4	61.3	60.9	52.8
FutureNav-8B, full data	10,435K (reported)	64.3	60.3	63.9	54.8
NaviWAM	1.38M (NF-only)	59.8	54.7	63.6	52.5

6.3.4 Stage-II Video Adaptation and Source Replay

Stage II follows the offline target-scene adaptation protocol in the main text. Target-scene video–instruction pairs are collected along routes independent of the evaluation routes and used for adaptation before evaluation on held-out routes.

The adaptation mixture contains retained source-navigation samples, Internet video–instruction pairs, and target-scene Isaac video–instruction pairs. Source-navigation replay supplies Policy supervision and associated NF targets. Internet and target-scene videos supply NF targets only, following equation 9. In the target-scene composition study, Isaac videos replace part of the Internet-video mixture at a fixed total video volume. Policy supervision continues to come from the source-navigation corpus.

6.4 Additional Results and Comparison Scope

6.4.1 Target-Scene Video Mixtures

Table 10 reports the full results for the four target-scene video proportions in Figure 3(b): 0%, 20%, 30%, and 40%. These configurations vary video composition at a fixed total video volume, with target-scene videos providing NF supervision.

All four metrics increase across the reported settings. Relative to the zero-target-video reference, the 40% configuration improves fine-grained SR/SPL by 6.91/11.88 percentage points and coarse-grained SR/SPL by 7.06/6.04 percentage points. These results describe navigation performance after NF adaptation with source Policy replay.

6.4.2 Waypoint Supervision and Additional Training Data

The main VLN-CE comparison uses the VLN-CE-only FutureNav configurations reported by Zhang et al. (2026), which are trained without additional navigation data. NaviWAM also obtains its Policy targets from source VLN-CE, while its additional video samples supply NF supervision only.

Table 11 includes FutureNav's full-data configurations for completeness. The FutureNav results and external-data counts are reproduced from the cited paper. For NaviWAM, the additional data count refers to processed video–instruction samples used for NF.

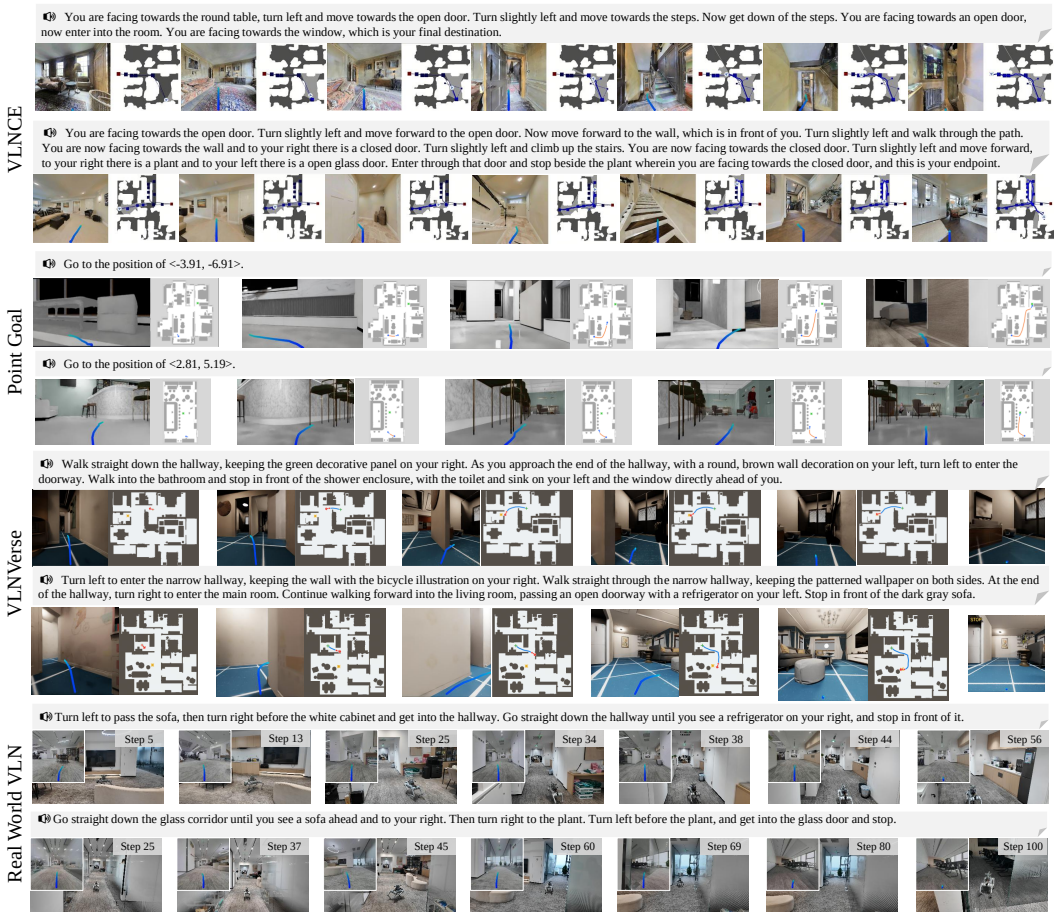

Figure 8: More qualitative examples across environments and benchmarks.

6.4.3 Simulation and Real-World Examples

Figure 8 presents qualitative navigation examples across simulated benchmarks and real-world environments. In the real-world examples, RGB observations are transmitted to a remote RTX 4090 for inference, and the onboard controller executes the predicted waypoints. The examples illustrate the task-to-trajectory interface under different instructions and goal specifications.

6.5 Future Work

Future work will examine the respective contributions of the shared WAM and trainable VLM conditioning pathways to transfer from NF supervision, together with the effect of patch-level versus pooled prediction targets. Another direction is to study Stage-II replay schedules and video-source composition under matched replay and optimization budgets, particularly for grounding coarse-grained instructions. Extending video-based adaptation to diverse human-recorded walkthroughs and conducting systematic real-world evaluations will further characterize robustness, long-horizon performance, and deployment latency.